Sample JSON Formats for Creating and Modifying Rules Through REST API

September 2020

This article contains sample JSON formats for POST and PUT requests for Risk Service rules.

**URL to create rules:** https://rba_ui_url:ui_port/risk/config/api/v1/<tenant_id>/rules

The following fields are mandatory for creating any Risk Service rule:

<table>
<thead>
<tr>
<th>JSON Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>A risk policy evaluates a rule only when it is enabled.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A unique name for the rule.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>The description of the rule evaluation condition.</td>
</tr>
</tbody>
</table>

**In this Article**

- External Parameters Rule JSON
- Cookie Rule JSON
- HTTP Header Rule JSON
- IP Address Rule JSON
- User Last Login Rule JSON
- User Time of Login Rule JSON
External Parameters Rule JSON

```json
{
    "name": "externalparamrule",
    "description": "Sample external param rule",
    "enabled": true,
    "externalParamConfigRule": [
        {
            "conditionGroup": [
                {
                    "condition": [
                        {
                            "conditionName": "OS",
                            "conditionValue": "win",
                            "contains": true
                        },
                        {
                            "conditionName": "patchlevel",
                            "lowerThreshold": "45",
                            "lessThanOrEqual": true
                        }
                    ],
                    "nextGroupCondition": "OR",
                    "operation": "AND",
                    "priority": 0
                },
                {
                    "condition": [
                        {
                            "conditionName": "gradelevel",
                            "lowerThreshold": "3",
                            "higherThreshold": "8",
                            "lessThan": true,
                            "greaterThan": true
                        }
                    ],
                    "nextGroupCondition": "OR",
                    "operation": "AND",
                    "priority": 0
                }
            ],
            "fetchFromParamSource": true,
            "negateResult": false,
            "paramSource": [
                {
                    "authenticationType": "None",
                    "dataFormat": "JSON",
                    "requestMethod": "POST",
                    "requestParameter": [
                        {
                            "name": "testparam",
                            "staticValue": "teststring",
                            "contextValue": null
                        }
                    ],
                    "requestTimeout": 30000,
                    "url": "http://external.site.com/rest/user/encodedUserId"
                }
            ]
        }
    ]
}
```
<table>
<thead>
<tr>
<th>JSON Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conditionGroup</td>
<td>Collection of conditions to evaluate.</td>
</tr>
<tr>
<td>operation</td>
<td>Specify how multiple conditions in a condition group should be combined. For example, for ConditionGroup1, evaluate using Condition1 AND Condition2. In this sample, OS contains win AND patchLevel &lt;= 45. Valid values are AND and OR.</td>
</tr>
<tr>
<td>nextGroupCondition</td>
<td>Specify how the condition groups should be combined. For example, ConditionGroup1 AND ConditionGroup2. Valid values are AND and OR.</td>
</tr>
<tr>
<td>priority</td>
<td>Specify the order of evaluating condition groups.</td>
</tr>
</tbody>
</table>

**Condition Definition**

<table>
<thead>
<tr>
<th>conditionName</th>
<th>Set to true to enable the comparison operator. Set only one operator to true. In this sample, OS contains win. The only exception is for performing the In Between operation. For example, gradeLevel in between 3 and 8 as shown in the sample.</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>Set this field if one of the following operators is true:</td>
</tr>
<tr>
<td></td>
<td>• contains</td>
</tr>
<tr>
<td></td>
<td>• doesNotContain</td>
</tr>
<tr>
<td></td>
<td>• equal</td>
</tr>
<tr>
<td></td>
<td>• equalIgnoreCase</td>
</tr>
<tr>
<td></td>
<td>• greaterThan</td>
</tr>
<tr>
<td></td>
<td>• greaterThanOrEqual</td>
</tr>
<tr>
<td></td>
<td>• lessThan</td>
</tr>
<tr>
<td></td>
<td>• lessThanOrEqual</td>
</tr>
<tr>
<td></td>
<td>• notEqual</td>
</tr>
<tr>
<td></td>
<td>• notEqualIgnoreCase</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>conditionValue</th>
<th>Set this field if one of the following operators is true:</th>
</tr>
</thead>
<tbody>
<tr>
<td>lowerThreshold</td>
<td>Set these fields if one of the following operators is true:</td>
</tr>
<tr>
<td>higherThreshold</td>
<td>• greaterThan</td>
</tr>
<tr>
<td></td>
<td>• greaterThanOrEqual</td>
</tr>
<tr>
<td></td>
<td>• lessThan</td>
</tr>
<tr>
<td></td>
<td>• lessThanOrEqual</td>
</tr>
</tbody>
</table>

**Fetch from External source**

| fetchFromParamSource | Set to true to get the data from an external source. This must be enabled for Risk service.                                   |
Sample JSON Formats for Creating and Modifying Rules Through REST API

**Cookie Rule JSON**

All fields are mandatory.

```
{
  "enabled": true,
  "name": "IntranetCookieRule",
  "description": "Verify that the request has the defined cookie",
  "knownCookieRule": [
    {
      "cookieName": "cname",
      "cookieValue": "cvalue",
      "negateResult": true,
      "autoCreateCookie": true,
      "cookieMaxAge": "5",
      "cookiePath": "/test",
      "cookieSecure": "false"
    }
  ]
}
```
<table>
<thead>
<tr>
<th>JSON Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cookieName</td>
<td>Specify the name of the cookie to be checked.</td>
</tr>
<tr>
<td>cookieValue</td>
<td>Specify the value that the cookie must contain.</td>
</tr>
<tr>
<td>negateResult</td>
<td>Specify true to make the rule succeed if the cookie does not contain the specified value.</td>
</tr>
<tr>
<td>autoCreateCookie</td>
<td>Specify true to enable creating the cookie automatically if the user authenticates successfully.</td>
</tr>
<tr>
<td>cookieMaxAge</td>
<td>Specify the validity of the cookie in days. The default value is one day.</td>
</tr>
<tr>
<td></td>
<td>This is required when autoCreateCookie is set to true.</td>
</tr>
<tr>
<td>cookiePath</td>
<td>Specify the path for the cookie. This is used while auto-creating the cookie.</td>
</tr>
<tr>
<td>cookieSecure</td>
<td>Specify true if the auto-created cookie should be secure.</td>
</tr>
</tbody>
</table>

**HTTP Header Rule JSON**

```json
{
    "enabled": "true",
    "name": "DeptHeaderRule",
    "description": "Validates if the request contains the finance department header",
    "httpheaderRule": [
        {
            "headerNames": [
                {
                    "value": "DEPARTMENT_HEADER"
                }
            ],
            "headerCondition": [
                {
                    "value": "finance"
                }
            ],
            "contains": true,
            "equals": false,
            "negateResult": false
        }
    ]
}
```
**IP Address Rule JSON**

```json
{
  "enabled": "true",
  "name": "InternalNetworkRule",
  "description": "Validates if a user is logging in using the corporate network",
  "ipaddressRule": [
    {
      "ipvalue": "1.1.1.1",
      "iprange": "2.2.2.2-3.3.3.3",
      "ipsubnet": "198.51.100.0/24",
      "iplistURL": null,
      "iplistURLConnectionTimeout": 10,
      "iplistURLUpdateInterval": 300,
      "considerHistoricalData": "false",
      "negateResult": "false"
    }
  ]
}
```

Specify the fields in the **Manual IP list** section or in the **Consuming from another source** section.

### JSON Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>headerNames &gt; value</td>
<td>Specify the name of the header to check.</td>
</tr>
<tr>
<td>headerCondition &gt; value</td>
<td>Specify the value of the header to check.</td>
</tr>
<tr>
<td>equals</td>
<td>Set any one of the following parameters to true:</td>
</tr>
<tr>
<td>contains</td>
<td>• equals: The rule succeeds if the value read from the header matches with the value specified for headerCondition.</td>
</tr>
<tr>
<td></td>
<td>• contains: The rule succeeds if the value read from the header contains the value specified for headerCondition.</td>
</tr>
<tr>
<td>negateResult</td>
<td>Use this to handle negative use cases when comparing the header value with the value specified for headerCondition.</td>
</tr>
<tr>
<td></td>
<td>For example, to evaluate when the value is not equal to the headerCondition value, set equals: true and negateResult: true</td>
</tr>
<tr>
<td></td>
<td>To evaluate when the value does not contain the headerCondition value, set contains: true and negateResult: true</td>
</tr>
<tr>
<td>considerHistoricalData</td>
<td>Specify true or false. When set to true, it checks the IP address in the user’s login history recorded in the database.</td>
</tr>
</tbody>
</table>

**Manually Providing the IP Address**

The default value is false and the rule succeeds if a user’s IP address is in the specified list. To block users with IP addresses in the specified list, set negateResult: true.
User Last Login Rule JSON

```json
{
   "enabled": "true",
   "name": "LastLoginRule",
   "description": "Validates the last successful login of the user",
   "lastLoginCookieRule": [
      {
         "cookieMaxAge": "5",
         "cookieName": "cookieName",
         "cookiePath": "/cookiePath",
         "cookieSecure": true,
         "cryptoKey": "cryptoKey@1234",
         "lastLoginAllowedAge": "3",
         "negateResult": false
      }
   ]
}
```

<table>
<thead>
<tr>
<th>JSON Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipvalue</td>
<td>Specify a comma separated list of IP addresses.</td>
</tr>
<tr>
<td>iprange</td>
<td>Specify a comma separated list of IP address ranges.</td>
</tr>
<tr>
<td>ipsubnet</td>
<td>Specify the list of IP subnets that must be allowed or blocked (depending on negateResult setting).</td>
</tr>
<tr>
<td>iplistURL</td>
<td>Specify the URL of the source that provides the list of IP addresses to check the IP address of users.</td>
</tr>
<tr>
<td>iplistURLConnectTimout</td>
<td>Specify the value in seconds. After this time, an unresponsive connection is closed.</td>
</tr>
<tr>
<td>iplistURLUpdateInterval</td>
<td>Specify the value in seconds. The connection will be refreshed at the specified interval.</td>
</tr>
<tr>
<td>cookieName</td>
<td>Specify a unique name for the cookie. This rule checks if the cookie exists by using this value and determine the risk accordingly. It will also create this cookie after the successful login.</td>
</tr>
<tr>
<td>cookieMaxAge</td>
<td>Specify the validity of the cookie in days.</td>
</tr>
<tr>
<td>cookiePath</td>
<td>Specify the URL to be used in the cookie.</td>
</tr>
<tr>
<td>cookieSecure</td>
<td>Specify true if you want the cookie to be secured by HTTPS. Allowed values are true and false.</td>
</tr>
<tr>
<td>cryptoKey</td>
<td>Specify the crypto key to encrypt the cookie.</td>
</tr>
<tr>
<td>lastLoginAllowedAge</td>
<td>Specify the number of days the cookie can be accessed from the same device or system. This value must be less than the value of cookieMaxAge.</td>
</tr>
</tbody>
</table>

Consuming Whitelist or Blacklist IP Addresses from Another Source

<table>
<thead>
<tr>
<th>JSON Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cookieName</td>
<td>Specify a unique name for the cookie. This rule checks if the cookie exists by using this value and determine the risk accordingly. It will also create this cookie after the successful login.</td>
</tr>
<tr>
<td>cookieMaxAge</td>
<td>Specify the validity of the cookie in days.</td>
</tr>
<tr>
<td>cookiePath</td>
<td>Specify the URL to be used in the cookie.</td>
</tr>
<tr>
<td>cookieSecure</td>
<td>Specify true if you want the cookie to be secured by HTTPS. Allowed values are true and false.</td>
</tr>
<tr>
<td>cryptoKey</td>
<td>Specify the crypto key to encrypt the cookie.</td>
</tr>
<tr>
<td>lastLoginAllowedAge</td>
<td>Specify the number of days the cookie can be accessed from the same device or system. This value must be less than the value of cookieMaxAge.</td>
</tr>
</tbody>
</table>
User Time of Login Rule JSON

```json
{
  "enabled": "true",
  "name": "TimeOfLoginRule",
  "description": "Validates if the user is logging in during business hours",
  "userTimeOfLoginRule": [
    {
      "considerHistoricalData": false,
      "negateResult": "false",
      "dayRange": [
        {
          "fromDay": "2",
          "toDay": "6"
        }
      ],
      "timeRange": [
        {
          "fromTime": "09:00:00",
          "toTime": "17:00:00"
        }
      ]
    }
  ]
}
```

<table>
<thead>
<tr>
<th>JSON Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fromDay</td>
<td>Specify the work week: Sunday (1) to Saturday (7).</td>
</tr>
<tr>
<td>toDay</td>
<td>In this example, it is Monday (2) to Friday (6).</td>
</tr>
<tr>
<td>fromTime</td>
<td>Specify the working hours in a day. In this example, it is 9 AM to 5PM.</td>
</tr>
<tr>
<td>toTime</td>
<td></td>
</tr>
<tr>
<td>considerHistoricalData</td>
<td>This is a boolean field to indicate if a user’s past login time must be considered as acceptable day and time while evaluating the rule.</td>
</tr>
<tr>
<td>negateResult</td>
<td>When you set it to true, the rule evaluates as true if the user is not logging in during the specified day and time range.</td>
</tr>
</tbody>
</table>

Legal Notice

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see [https://www.microfocus.com/about/legal/](https://www.microfocus.com/about/legal/).

© Copyright 2020 Micro Focus or one of its affiliates.